Process and Memory Management Commands on Cisco IOS-XR Software

This chapter describes the commands used to manage processes and memory on the Cisco IOS-XR software.

monitor processes

To display auto-updating statistics on processes in a full-screen mode, use the **monitor processes** command in EXEC mode.

monitor processes [**dumbtty**] [**location** *node-id*]

Syntax Description

dumbtty	(Optional) Displays the output of the command as if on a dumb terminal (the screen is not refreshed).
location *node-id*	(Optional) Displays the output of the command from the designated node. The *node-id* argument is entered in the rack/slot/module notation.

Defaults

If you omit all keywords, the command displays the top 10 processes of CPU usage for the local node, sorted in descending order by the time used. The display is cleared and updated every 5 seconds until you quit the **monitor processes** command by entering the **q** key.

Command Modes

EXEC

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

This command shows the top 10 processes of CPU usage. The display refreshes every 10 seconds.

- To change the parameters displayed by the **monitor processes** command, enter one of the key commands described in Table 14.
- To terminate the display and return to the system prompt, enter the **q** key.
- To list the interactive commands, type **?** during the display. Example:

The options are described in Table 14.

Table 14 Interactive Display Commands for the monitor processes Command

Command	Description
?	Display or print the interactive commands.
q	Quits the **monitor processes** display and returns to the system prompt.
n	Changes the number of processes to be displayed.
d	Changes the delay interval between updates.
l	Refreshes the screen.
t	Sorts display by time (default).

Table 14 Interactive Display Commands for the monitor processes Command (continued)

Command	Description
m	Sorts display by memory used.
c	Sorts display by number of open channels.
f	Sorts display by number of open files.

Examples

monitor processes

```
RP/0/RP1/CPU0:router# monitor processes

195 processes; 628 threads; 3300 channels, 4579 fds
CPU states: 47.6% idle, 1.2% user, 51.1% kernel
Memory: 2048M total, 1576M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  198     8    1      0    5:53:31 51.11% kernel
       52    5  215    44    5   228K    0:00:02  0.52% devc-conaux
      342    4  195    14    6     1M    0:00:08  0.34% wdsysmon
   495806    1    1    10    0   648K    0:00:00  0.16% ptop
      293    7   31    39   11   352K    0:00:09  0.07% shelfmgr
       55   11   24    14    5    16M    0:00:29  0.06% eth_server
      121    3   10     8    2   564K    0:00:05  0.02% bcm_process
      311    4    7    18    4   216K    0:00:02  0.01% sysdb_medusa_s
      138    4   14    40    5   240K    0:00:01  0.01% devc-vty
      265    5   31    19    4   204K    0:00:09  0.01% packet
```

monitor processes location

```
RP/0/RP1/CPU0:router# monitor processes location 0/rp1/cpu0

202 processes; 724 threads; 3750 channels, 5092 fds
CPU states: 48.8% idle, 0.8% user, 1.5% kernel
Memory: 2048M total, 1526M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  205     3    1      0   10:54:12  1.52% procnto-600-smp-cisco-in
str
      264    5   42    19    4   272K    0:00:15  0.37% packet
       53    2  202   564    0     1M    0:00:06  0.10% dllmgr
      180   15   93    42    6     1M    0:00:19  0.05% gsp
       69   22   94     8    3     1M    0:00:54  0.04% qnet
       67    5    4     6    0   956K    0:00:04  0.03% pkgfs
      156    2    6    18    1   480K    0:00:00  0.02% envmon
      294    1    6    12    1   112K    0:00:00  0.02% showd_lc
      314    3  185    14    4     1M    0:00:17  0.02% sysdb_svr_local
      310    4    7    18    4   276K    0:00:07  0.02% sysdb_medusa_s
```

monitor processes dumbtty

```
RP/0/RP1/CPU0:router# monitor processes dumbtty

Computing times...195 processes; 628 threads; 3721 channels, 4801 fds
CPU states: 37.1% idle, 1.1% user, 61.7% kernel
Memory: 2048M total, 1576M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  198     6    1      0    6:33:33 61.76% kernel
   544958    1    1     8    0   648K    0:00:00  0.64% ptop
      293    7   31    39   11   352K    0:00:10  0.10% shelfmgr
      180   15   82    42    6     5M    0:00:26  0.10% gsp
      304    3   14    29    7   304K    0:00:02  0.06% statsd_manager
       55   11   24    14    5    16M    0:00:32  0.03% eth_server
       70   22   91     8    3     1M    0:00:31  0.03% qnet
      153    2   35    18    4   120K    0:00:01  0.03% dsc
      303    3   25    34    5   292K    0:00:00  0.03% statsd_server
      121    3   10     8    2   564K    0:00:06  0.03% bcm_process

195 processes; 628 threads; 3409 channels, 4601 fds
CPU states: 46.5% idle, 0.5% user, 52.8% kernel
Memory: 2048M total, 1576M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  198     6    1      0    6:33:44 52.89% kernel
       52    5  215    44    5   228K    0:00:06  0.38% devc-conaux
      309    6   25    23    8   352K    0:00:08  0.03% sysdb_mc
      315    3  177    14    4     1M    0:00:12  0.03% sysdb_svr_local
      138    4   14    40    5   240K    0:00:02  0.02% devc-vty
      298    9   25   111    9     2M    0:00:09  0.01% snmpd
       67    4    4     7    0   804K    0:00:04  0.00% pkgfs
       53    2  195   547    0   944K    0:00:06  0.00% dllmgr
      311    4    7    18    4   216K    0:00:03  0.00% sysdb_medusa_s
      342    4  195    14    6     1M    0:00:08  0.00% wdsysmon
```

Use of Interactive Commands

When the **n** or **d** interactive command is used, the **monitor processes** command prompts you to enter a number. For example, when the interactive command **n** is entered, the prompt responds as shown below:

```
Enter number of procs to display: 15

195 processes; 628 threads; 3375 channels, 4495 fds
CPU states: 49.0% idle, 0.9% user, 50.0% kernel
Memory: 2048M total, 1576M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  198     2    1      0    6:11:43 50.01% kernel
       52    5  215    44    5   228K    0:00:05  0.72% devc-conaux
      293    7   31    39   11   352K    0:00:09  0.04% shelfmgr
      315    3  177    14    4     1M    0:00:11  0.03% sysdb_svr_local
      304    3   14    29    7   304K    0:00:01  0.02% statsd_manager
      309    6   25    23    8   352K    0:00:08  0.02% sysdb_mc
      342    4  195    14    6     1M    0:00:08  0.01% wdsysmon
      298    9   25   111    9     2M    0:00:09  0.00% snmpd
      265    5   31    19    4   204K    0:00:09  0.00% packet
      153    2   35    18    4   120K    0:00:00  0.00% dsc
      290    4    6    17    2   112K    0:00:00  0.00% sc_reddrv
      275    7   34    36    7   588K    0:00:00  0.00% qlink
      303    3   25    34    5   292K    0:00:00  0.00% statsd_server
      262    5   23    46    6     1M    0:00:00  0.00% ospf
      239    3   26    31    9   452K    0:00:00  0.00% lpts_pa
```

If the number you enter is outside the acceptable range, you will be prompted for another number:

```
Enter number of procs to display: 435
Please enter a number between 5 and 40
Enter number of procs to display:
```

Related Commands

Command	Description
monitor threads	Displays auto-updating thread statistics.
show processes	Displays information on all processes, or a single process.

monitor threads

To display auto-updating statistics on threads in a full-screen mode, use the **monitor threads** command in EXEC mode.

monitor threads [**dumbtty**] [**iteration** *number*] [**location** *node-id*]

Syntax Description

dumbtty	(Optional) Displays the output of the command as if on a dumb terminal (the screen is not refreshed).
iteration *number*	(Optional) Number of times the statistics display is to be updated, in the range from 0 to 4294967295.
location *node-id*	(Optional) Displays the output from the command from the designated node. The *node-id* argument is entered in the rack/slot/module notation.

Defaults

When all keywords are omitted, the command displays the first ten threads for the local node, sorted in descending order by the time used. The display is cleared and updated every 5 seconds until you quit the **monitor** command.

Command Modes

EXEC

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

This command shows the top 10 threads based on CPU usage. The display refreshes every 10 seconds.

- To change the parameters displayed by the **monitor threads** command, enter one of the key commands described in Table 15.
- To terminate the display and return to the system prompt, enter the **q** key.
- To list the interactive commands, type **?** during the display. Example:

The options are described in Table 15.

Table 15 Interactive Display Commands for the monitor threads Command

Command	Description
q	Quits the interactive display and returns to the system prompt.
n	Changes the number of threads to be displayed.

Table 15 Interactive Display Commands for the monitor threads Command (continued)

Command	Description
d	Changes the delay interval between updates.
l	Refreshes the screen.

Examples

monitor threads

```
RP/0/RP1/CPU0:router# monitor threads

195 processes; 628 threads;
CPU states: 98.2% idle, 0.9% user, 0.7% kernel
Memory: 2048M total, 1576M avail, page size 4K

   JID   TID LAST_CPU PRI STATE  HH:MM:SS      CPU  COMMAND
     1    12   1       10 Rcv     0:00:09     0.42% procnto-600-smp-cisco-instr
     1    25   1       10 Run     0:00:30     0.36% procnto-600-smp-cisco-instr
   342     1   1       19 Rcv     0:00:07     0.20% wdsysmon
    52     5   0       21 Rcv     0:00:03     0.15% devc-conaux
    52     3   1       18 Rcv     0:00:02     0.07% devc-conaux
532670     1   0       10 Rply    0:00:00     0.07% top
   293     6   0       55 Rcv     0:00:06     0.03% shelfmgr
    55     8   0       10 Rcv     0:00:02     0.03% eth_server
   315     3   0       10 Rcv     0:00:11     0.03% sysdb_svr_local
    55     7   0       55 Rcv     0:00:11     0.02% eth_server
```

monitor threads location

```
RP/0/RP1/CPU0:router# monitor threads location 0/rp1/cpu0

Computing times...195 processes; 628 threads;
CPU states: 95.1% idle, 2.7% user, 2.0% kernel
Memory: 2048M total, 1576M avail, page size 4K

   JID   TID LAST_CPU PRI STATE  HH:MM:SS      CPU  COMMAND
     1    25   0       10 Run     0:00:32     2.08% procnto-600-smp-cisco-instr
   265     5   0       10 SigW    0:00:09     0.89% packet
   279     1   1       10 Rcv     0:00:00     0.65% qsm
557246     1   0       10 Rply    0:00:00     0.51% top
   293     5   1       55 Rcv     0:00:01     0.07% shelfmgr
   180    13   1       10 Rcv     0:00:02     0.07% gsp
   315     3   0       10 Rcv     0:00:12     0.07% sysdb_svr_local
    55     7   1       55 Rcv     0:00:12     0.04% eth_server
   180     1   0       10 Rcv     0:00:01     0.04% gsp
   298     9   0       10 Rcv     0:00:01     0.04% snmpd
```

Use of Interactive Commands

When the **n** or **d** interactive command is used, the **monitor threads** command prompts for a number appropriate to the specific interactive command. The following is sample output from the **monitor threads** command showing the use of the interactive command **n** after the first display cycle to change the number of threads:

```
RP/0/RP1/CPU0:router# monitor threads

Computing times... 87 processes; 249 threads;
CPU states: 84.8% idle, 4.2% user, 10.9% kernel
Memory: 256M total, 175M avail, page size 4K

   JID   TID PRI STATE  HH:MM:SS      CPU  COMMAND
```

```
      1      6  10 Run      0:00:10    10.92% kernel
553049      1  10 Rply     0:00:00     4.20% top
    58      3  10 Rcv      0:00:24     0.00% sysdbsvr
     1      3  10 Rcv      0:00:21     0.00% kernel
    69      1  10 Rcv      0:00:20     0.00% wdsysmon
     1      5  10 Rcv      0:00:20     0.00% kernel
   159      2  10 Rcv      0:00:05     0.00% qnet
   160      1  10 Rcv      0:00:05     0.00% netio
   157      1  10 NSlp     0:00:04     0.00% envmon_periodic
   160      9  10 Intr     0:00:04     0.00% netio

n

Enter number of threads to display: 3
Please enter a number between 5 and 40
Enter number of threads to display: 8
87 processes; 249 threads;
CPU states: 95.3% idle, 2.9% user, 1.7% kernel
Memory: 256M total, 175M avail, page size 4K

   JID    TID  PRI STATE  HH:MM:SS      CPU  COMMAND
     1      6  10 Run      0:00:11     1.76% kernel
    69      1  10 Rcv      0:00:20     1.11% wdsysmon
    58      3  10 Rcv      0:00:24     0.40% sysdbsvr
   157      1  10 NSlp     0:00:04     0.23% envmon_periodic
   159     19  10 Rcv      0:00:02     0.20% qnet
553049      1  10 Rply     0:00:00     0.20% top
   159     12  10 Rcv      0:00:03     0.13% qnet
   160      1  10 Rcv      0:00:05     0.10% netio
```

When a number outside the acceptable range is entered, the acceptable range is displayed:

```
Please enter a number between 5 and 40
Enter number of threads to display:
```

Related Commands

Command	Description
monitor processes	Displays interactive, auto-updating process statistics in a full-screen mode.

process

To start, terminate, or restart a process, use the **process** command in EXEC mode.

process {**blocked** | **kill** | **restart** | **start**} [*executable-name* | *job-id*] [**location** {*node-id* | **all**}]

Syntax Description

blocked	Collects debug information. The node will be restarted if the mandatory reboot flag is set.
kill	Kills (stops) a process. The process will not be restarted even if considered “mandatory”.
restart	Restarts a process.
start	Starts a process.
executable-name	(Optional) Performs the action for all the simultaneously executing instances of the process, if applicable.
job-id	(Optional) Performs the action for only the process instance associated with the *job-id*.
location *node-id*	(Optional) Specifies a node. The *node-id* argument is entered in the rack/slot/module notation.
location all	(Optional) Specifies all nodes.

Defaults

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

Under normal circumstances, processes are started and restarted automatically by the operating system as required. If a process crashes, it will be automatically restarted.

You can also use the **process** commands to manually stop, start or restart individual processes. These commands should be used only under the supervision of a Cisco support representative because they can cause a loss of router operations.

Caution Manually killing or restarting a process can seriously impact the operation of a router. Use these commands only under the direction of a Cisco technical support representative.

process kill

The **process kill** command will shut down (terminate) the specified process and copies associated with the specified process. The process is not restarted, even if considered “mandatory.” Use the **show processes** command to display a list of executable processes running on the system.

Caution Killing a process can result in an RP failover, system failure or both. This command is intended for use only under the direct supervision of a Cisco technical support representative.

process restart

The **process restart** command is used to restart a process, such as a process that is not functioning optimally.

process start

The **process start** command starts a process that is not currently running, such as a process that was terminated using the **process kill** command. If multiple copies are on the system, all instances of the process will be started simultaneously.

process blocked

This command is used by Cisco support representatives to collect debug information about a process. If the mandatory command is set for a process, the **process blocked** command will also cause the node to restart.

Examples

In the following example, the OSPF process is restarted with the **process restart**. command:

```
RP/0/RP1/CPU0:router# process restart isis
RP/0/RP1/CPU0:router#RP/0/RP1/CPU0:Mar 30 15:24:41 : isis[343]: %ISIS-6-INFO_ST
RTUP_START : Cisco NSF controlled start beginning
RP/0/RP1/CPU0:router#RP/0/RP1/CPU0:Mar 30 15:24:52 : isis[352]: %ISIS-6-INFO_ST
RTUP_FINISH : Cold controlled start completed
```

In the following example, the OSPF process is terminated with the **process kill** command:

```
RP/0/RP1/CPU0:router# process kill isis
RP/0/RP1/CPU0:router#
```

In the following example, the OSPF process is started with the **process start** command:

```
RP/0/RP1/CPU0:router# process start isis
RP/0/RP1/CPU0:router#RP/0/RP1/CPU0:Mar 30 15:27:19 : isis[227]: %ISIS-6-INFO_STA
RTUP_START : Cold controlled start beginning
RP/0/RP1/CPU0:Mar 30 15:27:31 : isis[352]: %ISIS-6-INFO_STARTUP_FINISH : Cold co
ntrolled start completed
```

Related Commands

Command	Description
process mandatory	Sets the options for mandatory processes.
show processes	Displays information on the running processes.

process core

To change the core dump options for a process, use the **process core** command in EXEC mode.

process {*executable-name* | *job-id*} **core** {**mainmem** | **mainmem-sharedmem** | **mainmem-text** | **mainmem-text-sharedmem** | **sharedmem** | **copy** | **off**} [**maxcore** *value*] [**location** *node-id*]

Syntax Description

executable-name	Executable name of the process for which you want to change core dump options. Specifying an *executable-name* value changes the core dumping option for multiple instances of a running process.
job-id	Job ID associated with the process instance. Specifying a *job-id* value changes the core dumping option for only a single instance of a running process.
core	Indicates a core dump option change for the specified process.
mainmem	Dumps the main memory of a crashed process.
mainmem-sharedmem	Dumps the main memory and the shared memory of a crashed process.
mainmem-text	Dumps the main memory and text of a crashed process.
mainmem-text-sharedmem	Dumps the main memory, shared memory and text of a crashed process.
sharedmem	Dumps the shared memory of a crashed process.
copy	Copies a core dump locally before performing the core dump.
off	Indicates that a core dump will not be taken upon the termination of the specified process.
maxcore *value*	(Optional) Maximum number of core dumps allowed for the specified process since its creation.
location *node-id*	(Optional) Sets the core dump options to a designated node. The *node-id* argument is entered in the rack/slot/module notation.

Defaults

By default, processes are configured to dump shared memory, text area, stack, data section and heap.

Command Modes

EXEC

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

The modular architecture of the Cisco IOS-XR software allows core dumps for individual processes. By default, processes are configured to dump shared memory, text area, stack, data section and heap.

Specifying the process name (*executable-name*) changes the core dumping option for all instances of the process. Specifying a *job-id* value changes the core dumping option for a single instance of a running process.

Examples

In the following example, the **process core** command is used to enable dumping of main memory and shared memory:

```
RP/0/RP1/CPU0:router# process ospf core mainmem-sharedmem
```

In the following example, the **process core** command is used to turn off core dumping for a process:

```
RP/0/RP1/CPU0:router# process media_ether_config_di core off
```

Core dumping is turned back on by specifying the type of core dump for a process, as shown in the following example:

```
RP/0/RP1/CPU0:router# process media_ether_config_di core mainmem-text-sharedmem
```

Related Commands

Command	Description
show processes	Displays information about processes.

process mandatory

To set the mandatory reboot options for a process, use the **process mandatory** command in EXEC mode.

process mandatory

process mandatory {on | off} {*executable-name* | *job-id***}** [**location** *node-id*]

process mandatory reboot

process mandatory reboot {enable | disable}

process mandatory toggle

process mandatory toggle {*executable-name* | *job-id***} [location** *node-id***]**

Syntax Description

on	Turns on mandatory process attribute.
off	Turns off the mandatory process attribute. The process will not be considered mandatory.
reboot	Enables or disables the reboot action when mandatory process fails.
toggle	Toggles mandatory process attribute
executable-name	Executable name of the process you want to terminate. Terminates the process and all the simultaneously executing copies, if applicable.
job-id	Job ID associated with the process you want to terminate. Terminates only the process associated with the job ID.
location *node-id*	(Optional) Sets the mandatory settings for a process on a designated node. The *node-id* argument is entered in the rack/slot/module notation.

Defaults

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

If a process unexpectedly goes down, the following action will occur based on whether the process is considered “mandatory.”

- If the process is mandatory, and the process cannot be restarted, the node will automatically reboot.

- If the process is not mandatory and cannot be restarted, it will stay down and the node will not reboot.

Examples

Turning the mandatory attribute on or off for a process

In the following example, the mandatory attribute is turned on for the “media_ether_config_di” process:

```
RP/0/RP1/CPU0:router# process mandatory on media_ether_config_di
```

Turning the reboot option on or off with the process mandatory reboot command

In the following example, the system is set to reboot the node if a mandatory process goes down and cannot be restarted:

```
RP/0/RP1/CPU0:router# process mandatory reboot enable
RP/0/RP1/CPU0:Mar 19 19:28:10 : sysmgr[71]: %SYSMGR-4-MANDATORY_REBOOT_ENABLE :
mandatory reboot option enabled by request
```

In the following example, the system is set *not* to reboot the node if a mandatory process goes down and cannot be restarted. In this case, the mandatory process will be restarted, but the node will not ne rebooted.

```
RP/0/RP1/CPU0:router# process mandatory reboot disable
RP/0/RP1/CPU0:Mar 19 19:31:20 : sysmgr[71]: %SYSMGR-4-MANDATORY_REBOOT_OVERRIDE
: mandatory reboot option overridden by request
```

Related Commands

Command	Description
show processes	Displays information, attributes and settings for a processes.

show memory

To display the available physical memory on the networking device and the memory usage of processes on the networking device, use the **show memory** command in EXEC mode.

show memory [*job-id* | **location** *node-id* | **summary**]

Syntax Description

job-id	(Optional) Displays the memory available and memory usage information for only the process associated with this job identifier. If the *job-id* argument is not specified, this command displays information for all running processes.
location *node-id*	(Optional) Displays the available physical memory from the designated node. The *node-id* argument is entered in the rack/slot/module notation.
summary	(Optional) Displays only a summary of the physical memory and memory usage information.

Command Modes

EXEC

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

Examples

The following is partial sample output from the **show memory** command entered without keywords or arguments:

```
RP/0/RP1/CPU0:router# show memory

Physical Memory:2048M total
 Application Memory :1802M (1636M available)
 Image:116M (bootram:116M)
 Reserved:128M, IOMem:0, flashfsys:0
 Total shared window:0

kernel:jid 1
Address         Bytes           What
0008f000        12288           Program Stack
000b2000        12288           Program Stack
Total Allocated Memory:0
Total Shared Memory:0

sbin/devc-pty:jid 68
Address         Bytes           What
4817f000        4096            Program Stack (pages not allocated)
48180000        516096          Program Stack (pages not allocated)
481fe000        8192            Program Stack
48200000        28672           Physical Mapped Memory
```

```
48207000        4096            ANON FIXED ELF SYSRAM
48208000        4096            ANON FIXED ELF SYSRAM
```

The following is sample output from the **show memory** command entered with the job ID 7 to show the memory usage information for the process associated with this job identifier:

```
RP/0/RP1/CPU0:router# show memory 7

Physical Memory: 256M total
 Application Memory : 249M (217M available)
 Image: 2M (bootram: 2M)
 Reserved: 4M, IOMem: 0, flashfsys: 0

sbin/pipe: jid 7
Address         Bytes           What
07f7c000        126976          Program Stack (pages not allocated)
07f9b000        4096            Program Stack
07f9d000        126976          Program Stack (pages not allocated)
07fbc000        4096            Program Stack
07fbe000        126976          Program Stack (pages not allocated)
07fdd000        4096            Program Stack
07fdf000        126976          Program Stack (pages not allocated)
07ffe000        4096            Program Stack
08000000        122880          Program Stack (pages not allocated)
0801e000        8192            Program Stack
08020000        12288           Physical Mapped Memory
08023000        4096            Program Text or Data
08024000        4096            Program Text or Data
08025000        16384           Allocated Memory
08029000        16384           Allocated Memory
7c001000        319488          DLL Text libc.dll
7e000000        8192            DLL Data libc.dll
```

Related Commands

Command	Description
show memory heap	Displays information about the heap space for a process.
show processes	Displays information about processes, including memory usage.

show memory heap

To display information about the heap space for a process, use the **show memory heap** command in EXEC mode.

show memory heap [**allocated**] [**dllname**] [**failure**] [**free**] [**summary**] *job-id*

Syntax Description

allocated	(Optional) Displays a list of all allocated heap blocks.
dllname	(Optional) Displays heaps with DLL names.
failure	(Optional) Displays a summary of heap failures.
free	(Optional) Displays a list of all free heap blocks.
summary	(Optional) Displays a summary of the information about the heap space.
job-id	Information for the process associated with this job identifier.

Command Modes

EXEC

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

Examples

The following is sample output from the **show memory heap** command, specifying only the job identifier 111:

```
RP/0/RP1/CPU0:router> show memory heap 111

Malloc summary for pid 16433:
  Heapsize 16384: allocd 6328, free 8820, overhead 1236
  Calls: mallocs 144; reallocs 73; frees 5; [core-allocs 1; core-frees 0]
Block Allocated List
Total      Total      Block       Name/ID/Caller
Usize      Size       Count

0x000008c1 0x000008cc 0x00000001  0x7c018a10
0x000005ac 0x00000974 0x00000079  0x7c02b9e0
0x000004f0 0x000004f8 0x00000001  0x7c02b6fc
0x00000080 0x00000088 0x00000001  0x7c01936c
0x00000034 0x00000048 0x00000001  0x7c018954
0x00000024 0x00000030 0x00000001  0x7c019278
0x00000018 0x00000020 0x00000001  0x7c019b2c
0x00000008 0x00000010 0x00000001  0x7c017178
0x00000008 0x00000010 0x00000001  0x7c00fb54
0x00000008 0x00000010 0x00000001  0x7c00fb80
0x00000008 0x00000010 0x00000001  0x7c00fbb8
```

The following is sample output from the **show memory heap** command, specifying the **allocated** and **free** keywords and the job identifier:

```
RP/0/RP1/CPU0:router# show memory heap allocated free 5

Block Allocated List
Usize      Size       Address    Name/ID/Caller

0x000008c1 0x000008cc 0x08029e7c 0x7c018a10
0x000004fc 0x00000504 0x08029554[0x18 0x30-byte objects: 0x1 allocd, 0x17 free]
 caller(id)/usize/addr: 0x7c019278/0x28/0x8029574
0x000004f0 0x000004f8 0x0802bc00 0x7c02b6fc
0x0000037c 0x00000384 0x08029a60[0x18 0x20-byte objects: 0x1 allocd, 0x17 free]
 caller(id)/usize/addr: 0x7c019b2c/0x1c/0x8029a80
0x0000031c 0x00000324 0x0802a7d0[0x20 0x14-byte objects: 0x20 allocd, 0x0 free]
 caller(id)/usize/addr: 0x7c02b9e0/0x10/0x802a7f0,0x802a808,0x802a820,0x802a838,
0x802a850,0x802a868,0x802a880,0x802a898,0x802a8b0,0x802a8c8,0x802a8e0,0x802a8f8,
0x802a910,0x802a928,0x802a940,0x802a958,0x802a970,0x802a988,0x802a9a0,0x802a9b8,
0x802a9d0,0x802a9e8,0x802aa00,0x802aa18,0x802aa30,0x802aa48,0x802aa60,0x802aa78,
0x802aa90,0x802aaa8,0x802aac0,0x802aad8
0x0000031c 0x00000324 0x0802ac2c[0x20 0x14-byte objects: 0x20 allocd, 0x0 free]
 caller(id)/usize/addr: 0x7c02b9e0/0x10/0x802ac4c,0x802ac64,0x802ac7c,0x802ac94,
0x802acac,0x802acc4,0x802acdc,0x802acf4,0x802ad0c,0x802ad24,0x802ad3c,0x802ad54,
0x802ad6c,0x802ad84,0x802ad9c,0x802adb4,0x802adcc,0x802ade4,0x802adfc,0x802ae14,
0x802ae2c,0x802ae44,0x802ae5c,0x802ae74,0x802ae8c,0x802aea4,0x802aebc,0x802aed4,
0x802aeec,0x802af04,0x802af1c,0x802af34
0x0000031c 0x00000324 0x0802b1a8[0x20 0x14-byte objects: 0x20 allocd, 0x0 free]
 caller(id)/usize/addr: 0x7c02b9e0/0x10/0x802b1c8,0x802b1e0,0x802b1f8,0x802b210,
0x802b228,0x802b240,0x802b258,0x802b270,0x802b288,0x802b2a0,0x802b2b8,0x802b2d0,
0x802b2e8,0x802b300,0x802b318,0x802b330,0x802b348,0x802b360,0x802b378,0x802b390,
0x802b3a8,0x802b3c0,0x802b3d8,0x802b3f0,0x802b408,0x802b420,0x802b438,0x802b450,
0x802b468,0x802b480,0x802b498,0x802b4b0
0x0000031c 0x00000324 0x0802b8d4[0x20 0x14-byte objects: 0x19 allocd, 0x7 free]
 caller(id)/usize/addr: 0x7c02b9e0/0x10/0x802b8f4,0x802b90c,0x802b924,0x802b93c,
0x802b954,0x802b96c,0x802b984,0x802b99c,0x802b9b4,0x802b9cc,0x802b9e4,0x802b9fc,
0x802ba14,0x802ba2c,0x802ba44,0x802ba5c,0x802ba74,0x802ba8c,0x802baa4,0x802babc,
0x802bad4,0x802baec,0x802bb04,0x802bb1c,0x802bb34
0x0000029c 0x000002a4 0x0802901c[0x20 0x10-byte objects: 0x4 allocd, 0x1c free]
 caller(id)/usize/addr: 0x7c017178/0xc/0x802903c 0x7c00fb54/0xc/0x8029050 0x7c00
fb80/0xc/0x8029064 0x7c00fbb8/0xc/0x8029078
0x0000027c 0x00000284 0x080292c8[0x8 0x48-byte objects: 0x1 allocd, 0x7 free]
 caller(id)/usize/addr: 0x7c018954/0x38/0x80292e8
0x00000080 0x00000088 0x08029dec 0x7c01936c
Block Free List
Size       Address
0x00000078 0x0802a750
0x00000128 0x0802aafc
0x00000248 0x0802af58
0x000003f8 0x0802b4d4
0x00000ef8 0x0802c100
```

The following is sample output from the **show memory heap** command, specifying the **summary** keyword and the job identifier:

```
RP/0/RP1/CPU0:router# show memory heap summary 65

Malloc summary for pid 20495 process pcmciad:
  Heapsize 65536: allocd 40332, free 16568, overhead 8636
  Calls: mallocs 883; reallocs 3; frees 671; [core-allocs 4; core-frees 0]
Band size 16, element per block 48, nbuint 1
  Completely free blocks: 0
  Block alloced: 2, Block freed: 0
  allocs: 85, frees: 20
  allocmem: 1040, freemem: 496, overhead: 448
  blocks: 2, blknodes: 96
Band size 24, element per block 34, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 243, frees: 223
  allocmem: 480, freemem: 336, overhead: 168
  blocks: 1, blknodes: 34
Band size 32, element per block 26, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 107, frees: 97
  allocmem: 320, freemem: 512, overhead: 136
  blocks: 1, blknodes: 26
Band size 40, element per block 22, nbuint 1
  Completely free blocks: 0
  Block alloced: 2, Block freed: 0
  allocs: 98, frees: 74
  allocmem: 960, freemem: 800, overhead: 240
  blocks: 2, blknodes: 44
Band size 48, element per block 18, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 53, frees: 42
  allocmem: 528, freemem: 336, overhead: 104
  blocks: 1, blknodes: 18
Band size 56, element per block 16, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 8, frees: 4
  allocmem: 224, freemem: 672, overhead: 96
  blocks: 1, blknodes: 16
Band size 64, element per block 14, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 6, frees: 2
  allocmem: 256, freemem: 640, overhead: 88
  blocks: 1, blknodes: 14
Band size 72, element per block 12, nbuint 1
  Completely free blocks: 0
  Block alloced: 1, Block freed: 0
  allocs: 1, frees: 0
  allocmem: 72, freemem: 792, overhead: 80
  blocks: 1, blknodes: 12
```

Related Commands

Command	Description
show memory	Displays the available physical memory and processes memory on a router.

show processes

To display information about active processes, use the **show processes** command in EXEC mode.

show processes [*job-id* | *process-name*] {**aborts** | **blocked** | **boot** | **cpu** | **distribution** | **dynamic** | **failover** | **family** | **files** | **location** | **log** | **mandatory** | **memory** | **searchpath** | **signal** | **startup** | **threadname**} [**location** *node-id*]

Syntax Description

job-id	(Optional) Displays information for only the process instance associated with the *job-id* argument.
process-name	(Optional) Displays information for all the simultaneously executing instances of the process, if applicable.
aborts	Displays process aborts.
blocked	Displays details about reply/send/mutex blocked processes.
boot	Displays process boot information.
cpu	Displays CPU use per process.
distribution	Displays the distribution of processes.
dynamic	Displays process data for dynamically created processes.
failover	Displays process failover information.
family	Displays the process session and family information.
files	Displays information about open files and open communication channels.
log	Displays process log.
mandatory	Displays process data for mandatory processes.
memory	Displays information about the text, data, and stack usage for processes.
searchpath	Displays the search path.
signal	Displays the signal options for blocked, pending, ignored, and queued signals.
startup	Displays process data for processes created at startup
threadname	Displays thread names.
location *node-id*	(Optional) Displays information about the active processes from a designated node. The *node-id* argument is entered in the rack/slot/module notation.

Command Modes

EXEC

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

The **show processes** command displays general information on the active processes. To view more detailed information for a process, include the *job-id* or *process-name.*

You can also use the **monitor** command to determine the top users of the CPU.

Examples

The following is partial sample output from the **show processes** command:

```
RP/0/RP1/CPU0:router# show processes

JID   TID LastCPU Stack pri state     HR:MM:SS:MSEC NAME
1      1    0       0K   0 Ready         1:57:41:0542 procnto-600-smp-cisco-instr
1      2    1       0K   0 Running       1:57:37:0022 procnto-600-smp-cisco-instr
1      3    1       0K  10 Receive       0:00:05:0723 procnto-600-smp-cisco-instr
1      4    1       0K  10 Receive       0:00:00:0001 procnto-600-smp-cisco-instr
1      5    0       0K  63 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      6    1       0K  63 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      7    0       0K  63 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      8    0       0K  10 Receive       0:00:01:0885 procnto-600-smp-cisco-instr
1      9    1       0K  10 Receive       0:00:03:0416 procnto-600-smp-cisco-instr
1      10   1       0K  10 Receive       0:00:00:0001 procnto-600-smp-cisco-instr
1      11   1       0K  10 Receive       0:00:04:0861 procnto-600-smp-cisco-instr
1      15   0       0K  10 Receive       0:00:02:0020 procnto-600-smp-cisco-instr
1      18   1       0K  10 Receive       0:00:03:0278 procnto-600-smp-cisco-instr
1      20   1       0K  10 Receive       0:00:00:0732 procnto-600-smp-cisco-instr
1      21   1       0K  10 Receive       0:00:02:0692 procnto-600-smp-cisco-instr
1      22   0       0K  10 Running       0:00:03:0788 procnto-600-smp-cisco-instr
1      23   1       0K  10 Receive       0:00:11:0785 procnto-600-smp-cisco-instr
1      25   0       0K  10 Receive       0:00:04:0037 procnto-600-smp-cisco-instr
 --More--
```

The **show process** *process-name* command displays detailed information about a process:

```
RP/0/RP1/CPU0:router# show processes ospf

                  Job Id: 261
                     PID: 139453
         Executable path: /hfr-rout-0.44.0/bin/ospf
              Instance #: 1
              Version ID: 00.00.0000
                 Respawn: ON
           Respawn count: 1
  Max. spawns per minute: 12
            Last started: Wed Mar 17 07:46:26 2004
           Process state: Run
           Package state: Normal
       Started on config: cfg/gl/ipv4-ospf/proc/100/ord_a/routerid
                    core: TEXT SHAREDMEM MAINMEM
               Max. core: 0
               Mandatory: ON
               Placement: ON
            startup_path: /pkg/startup/ospf.startup
        Process cpu time: 0.410 user, 0.183 kernel, 0.593 total
JID    TID  LastCPU  Stack pri state        HR:MM:SS:MSEC NAME
261    1    0      40K  10 Receive       0:00:00:0397 ospf
261    2    1      40K  10 Receive       0:00:00:0003 ospf
261    3    0      40K  10 Receive       0:00:00:0007 ospf
261    4    1      40K  10 Condvar       0:00:00:0000 ospf
 --More--
```

The **show processes memory** command displays details of memory usage for a given process or for all processes, as shown in the following example:

```
RP/0/RP1/CPU0:router# show processes memory

JID    Text     Data     Stack    Dynamic  Process
55     28672    4096     69632    17072128 eth_server
164    143360   4096     20480    13238272 hfr_fgid_server
317    167936   4096     45056    10526720 syslogd
122    512000   4096     77824    9797632  bgp
265    57344    4096     57344    5877760  parser_server
254    40960    4096     143360   3084288  netio
63     8192     4096     24576    2314240  nvram
314    4096     4096     36864    1699840  sysdb_svr_local
341    495616   4096     40960    1576960  wdsysmon
259    53248    4096     28672    1490944  nvgen_server
189    32768    4096     32768    1425408  hd_drv
69     77824    4096     110592   1421312  qnet
348    323584   4096     40960    1392640  ospf
347    323584   4096     40960    1392640  ospf
346    323584   4096     40960    1392640  ospf
345    323584   4096     40960    1392640  ospf
344    323584   4096     40960    1392640  ospf
261    323584   4096     40960    1392640  ospf
 --More--
```

The following is partial sample output from the **show processes signal** command:

```
RP/0/RP1/CPU0:router# show processes signal

pid name                   signals pending  signals ignored   signals queued

        tid   signals blocked  signals pending
      1 kernel                0000000000000000 0000000006c20100 0000000000000000
        1    0000000000000000 00ffffffffffffff
        2    0000000000000000 00ffffffffffffff
        3    0000000000000000 00ffffffffbffeff
        4    0000000000000000 00ffffffffbffeff
        5    0000000000000000 00ffffffffbffeff
        6    0000000000000000 00ffffffffbffeff
        7    0000000000000000 00ffffffffbffeff
        8    0000000000000000 00ffffffffbffeff
        9    0000000000000000 00ffffffffbffeff
        10   0000000000000000 00ffffffffbffeff
        11   0000000000000000 00ffffffffbffeff
        12   0000000000000000 00ffffffffbffeff
        14   0000000000000000 00ffffffffbffeff
        14   0000000000000000 00ffffffffbffeff
        15   0000000000000000 00ffffffffbffeff
        16   0000000000000000 00ffffffffbffeff
 --More--
```

The following is partial sample output from the **show processes family** command:

```
RP/0/RP1/CPU0:router# show processes family

id name                         session  pgroup    ppid sibling   child
      1 kernel                             1       1       0       0      71
     72 wd-mbi                             1      72      71       5       0
     53 dllmgr                             1       2      71       0       0
    278 qsm                                1     278      71      54       0
     67 pkgfs                              1      67      71      72   65545
     68 devc-pty                           1      68      71      67       0
     52 devc-conaux                        1      52      71      68   65669
```

```
  65545 pkgfs                              1   65545      67       0       0
  65546 ksh                            65546   65546      52       0       0
     66 pipe                               1      66      71      52       0
     56 devf-scrp                          1      56      71      61       0
     61 mqueue                             1      61      71      66       0
     64 pci_bus_mgr                        1      64      71      56       0
     65 pcmciad                            1      65      71      64   65559
  65552 cardmgrd                           1      65      65       0       0
     70 syslogd_helper                     1      70      71      65       0
     63 nvram                              1      63      71      70       0
     55 eth_server                         1      55      71      63       0
 --More--
```

The following is partial sample output from the **show processes files** command:

```
RP/0/RP1/CPU0:router# show processes files

JID        Open Files  NAME
1          13          kernel
72         4           wd-mbi
53         588         dllmgr
278        16          qsm
67         6           pkgfs
68         4           devc-pty
52         45          devc-conaux
65545      5           pkgfs
65546      5           ksh
66         4           pipe
56         3           devf-scrp
61         4           mqueue
64         9           pci_bus_mgr
65         11          pcmciad
65552      6           cardmgrd
70         16          syslogd_helper
63         13          nvram
55         14          eth_server
 --More--
```

The following is partial sample output from the **show processes blocked** command:

```
RP/0/RP1/CPU0:router# show processes blocked

    Jid     Pid Tid                Name State Blocked-on
  65546    4106   1                 ksh Reply    4104  devc-conaux
    105   41012   2             attachd Reply   20499  eth_server
    105   41012   3             attachd Reply    4109  mqueue
    324   41031   1         tftp_server Reply    4109  mqueue
  65669 1237125   1                exec Reply       1  kernel
    236  123014   2             lpts_fm Reply   41049  lpts_pa
    163  123022   2              fdiagd Reply   20499  eth_server
    163  123022   3              fdiagd Reply    4109  mqueue
    335  139436   1           udp_snmpd Reply  123005  udp
  65740 1401036   1                more Reply    4107  pipe
  65741 1401037   1      show_processes Reply       1  kernel
```

The following is partial sample output from the **show processes boot** command:

```
RP/0/RP1/CPU0:router# show processes boot

Band   Name    Finished  %Idle     JID     Ready    Last Process
----- -------- -------- -------- -------- ------- --------------------------
  1.0 MBI        15.582  67.770%       58  15.582 insthelper
 40.0 ARB        26.713  93.540%      281  11.131 redcon
100.0 INFRA     144.134  77.020%      198 117.421 instdir
150.0 ACTIVE    168.367   0.090%      271  24.233 policy_repository
```

```
999.0 FINAL     177.738   0.000%      172   9.371 fm_script_dir

Started Level    JID   Inst Ready             Process
------- ----- -------- ---- ------- ------------------------------
  0.000   0.5       72    1   0.000 wd-mbi
  0.000   1.0       53    1   0.000 dllmgr
  0.000   2.0       67    1   0.000 pkgfs
  0.000   3.0       52    1   0.000 devc-conaux
  0.000   3.0       68    1   0.000 devc-pty
  0.000   6.0       66    1   0.000 pipe
  0.000   8.0       61    1   0.000 mqueue
  0.000  16.0       56    1   0.000 devf-scrp
  0.000  20.0       64    1   0.000 pci_bus_mgr
 --More--
```

The following is sample output from the **show processes cpu** command:

```
RP/0/RP1/CPU0:router# show processes cpu

CPU utilization for one minute: 1%; five minutes: 1%; fifteen minutes: 1%

PID    1Min    5Min    15Min Process
1        0%      0%       0% kernel
3        0%      0%       0% dllmgr
4098     0%      0%       0% wd-mbi
4102     0%      0%       0% pkgfs
4103     0%      0%       0% devc-pty
4104     0%      0%       0% devc-conaux
4105     0%      0%       0% pkgfs
4106     0%      0%       0% ksh
4107     0%      0%       0% pipe
4109     0%      0%       0% mqueue
12300    0%      0%       0% devf-scrp
16398    0%      0%       0% pci_bus_mgr
20495    0%      0%       0% pcmciad
20496    0%      0%       0% cardmgrd
20497    0%      0%       0% syslogd_helper
 --More--
```

The following is partial sample output from the **show processes log** command:

```
RP/0/RP1/CPU0:router# show processes log

01/01 00:00:02.091 1 Hello from init !!
01/01 00:00:02.093 1 Wait for pkgfs at '/pkg'
01/01 00:00:03.138 1 Boot Device = /dev/null
01/01 00:00:03.139 1 Create event manager
01/01 00:00:03.199 1 Attach to msg channel
01/01 00:00:03.200 1 Create msg handling thread
01/01 00:00:03.200 2 sysmgr_lite_process_msg: In sysmgr_process_msg thread
01/01 00:00:03.200 2 Attaching respawn handler
01/01 00:00:03.201 1 read_init_startup_list: opening directory /pkg/init.d for .
init files
01/01 00:00:03.201 2 Attaching async handler
01/01 00:00:03.202 2 Attaching sync handler
01/01 00:00:03.202 2 starting ih_timer
01/01 00:00:03.202 2 lite_set_timer: id=1, 1800 seconds
01/01 00:00:03.202 2 Servicing msgs
01/01 00:00:03.205 1 read_init_startup_list: Opening /pkg/init.d/bfm.init
01/01 00:00:03.208 1 read_init_startup_list: finished /pkg/init.d/bfm.init pcb->
name=bfm_server
01/01 00:00:03.208 1 read_init_startup_list: Opening /pkg/init.d/clock_chip.init
01/01 00:00:03.210 1 read_init_startup_list: finished /pkg/init.d/clock_chip.ini
t pcb->name=clock_chip
```

```
01/01 00:00:03.211 1 read_init_startup_list: Opening /pkg/init.d/devc_conaux.ini
t
01/01 00:00:03.213 1 read_init_startup_list: finished /pkg/init.d/devc_conaux.in
it pcb->name=devc-conaux
01/01 00:00:03.213 1 read_init_startup_list: Opening /pkg/init.d/dllmgr.init
01/01 00:00:03.215 1 read_init_startup_list: finished /pkg/init.d/dllmgr.init pc
b->name=dllmgr
01/01 00:00:03.215 1 read_init_startup_list: Opening /pkg/init.d/dumper.init
 --More--
```

Related Commands

Command	Description
monitor processes	Displays auto-updating process statistics in a full-screen mode.
monitor threads	Displays auto-updating process and thread statistics in a full-screen mode.